



Bescheinigung

Certificate

Attestation

Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein.

The attached documents are exact copies of the European patent application described on the following page, as originally filed.

Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante.

Patentanmeldung Nr.

Patent application No.

Demande de brevet n°

00103798.5 / EP00103798

The organization code and number of your priority application, to be used for filing abroad under the Paris Convention, is EP00103798

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

R.C. van Dijk



Anmeldung Nr.:
Application no.: 00103798.5
Demande no:

Anmeldetag:
Date of filing: 23.02.00
Date de dépôt:

Anmelder/Applicant(s)/Demandeur(s):

Tektronix Berlin GmbH & Co. KG
Wernerwerkdamm 5
13629 Berlin/DE

Bezeichnung der Erfindung/Title of the invention/Titre de l'invention:
(Falls die Bezeichnung der Erfindung nicht angegeben ist, siehe Beschreibung.
If no title is shown please refer to the description.
Si aucun titre n'est indiqué se référer à la description.)

Verfahren zum Erstellen eines Kommunikationsablaufs zwischen mindestens zwei Instanzen und
Protokolltester hierfür

In anspruch genommene Priorität(en) / Priority(ies) claimed / Priorité(s) revendiquée(s)
Staat/Tag/Aktenzeichen / State/Date/File no. / Pays/Date/Numéro de dépôt:

Internationale Patentklassifikation / International Patent Classification / Classification internationale de brevets:

H04Q

Am Anmeldetag benannte Vertragsstaaten / Contracting states designated at date of filing / Etats contractants désignées lors du dépôt:

DE FI FR GB IT

STRASSE & HOFSTETTER

**EPO - Munich
50**

23. Feb. 2000

**PATENTANWÄLTE
EUROPEAN PATENT ATTORNEYS
EUROPEAN TRADEMARK ATTORNEYS**

München

Dr.rer.nat. ALFONS J. HOFSTETTER ^{1,2}

Dr.-Ing. EDUARD F. SCHURACK ^{1,2}

Hannover

Dipl.-Ing. MICHAEL SKORA ^{1,2}

Dipl.-Ing. JOACHIM STRASSE (1962-1998) ^{1,2,3}

Dr. jur. PETER C. BITTNER ⁴

¹ Patentanwalt

² Zugelassener Vertreter beim Europäischen
Patentamt und Europäischen Harmonisierungs-
amt für den Binnenmarkt

³ Consultant / Nur in beratender Tätigkeit s. 1999

⁴ Consultant / Nur in beratender Tätigkeit s. 1997

In Zusammenarbeit mit / In cooperation with

LENZING GERBER

Patentanwälte · Düsseldorf

Anwaltsakte: 25380

**Tektronix, Inc.
26600 S.W. Parkway Avenue
P.O. Box 1000
Wilsonville, Oregon 97070-1000
U.S.A.**

EUROPÄISCHE PATENTNEUANMELDUNG

**Verfahren zum Erstellen eines Ablaufs einer zwischen
mindestens zwei Instanzen ablaufenden Kommunikation und dieses
Verfahren realisierender Protokolltester**

**BALANSTRASSE 57 TELEFON +49/(0)89 - 4509180
D-81541 MÜNCHEN FAX +49/(0)89 - 45091812**

**TIERGARTENSTRASSE 122
D-30559 HANNOVER**

**TELEFON +49/(0)511 - 5106365
Fax +49/(0)511 - 5106371**

EMAIL Strasse-Hofstetter@T-Online.de

23. Feb. 2000

Tektronix, Inc.
Europäische Patentneuanmeldung

Anwaltsakte: 25380

**Verfahren zum Erstellen eines Ablaufs einer zwischen
mindestens zwei Instanzen ablaufenden Kommunikation und dieses
Verfahren realisierender Protokolltester**

Beschreibung:

Die vorliegende Erfindung betrifft ein Verfahren zum Erstellen eines Ablaufs einer zwischen mindestens zwei Instanzen ablaufenden Kommunikation, wobei eine Instanz ein Protokolltester ist. Sie betrifft überdies einen Protokolltester, in dem dieses Verfahren realisiert ist.

Im Bereich des Protokolltests ist es nötig, den Kommunikationsablauf, durch den ein Test beschrieben wird, eindeutig zu spezifizieren, so daß dieser Ablauf automatisch maschinell ausgeführt werden kann. Sprachen, wie zum Beispiel TTCN (Tree and Tabular Combined Notation) ermöglichen dies, sind aber komplex und für einen ungeübten Leser schwer zu verstehen. TTCN hat sich im Bereich des Conformance Testing durchgesetzt, da diese Tests sehr umfangreich sind und TTCN solche umfangreichen Tests gut unterstützt. Daneben gibt es diverse proprietäre Testbeschreibungssprachen. Um die Verständlichkeit zu erleichtern, wird zu Dokumentations- und Beschreibungszwecken einfacher Abläufe die standardisierte Sprache MSC (Message Sequence Charts) verwendet. Detailliertere Angaben zu MSC können der ITU-T Z.120 entnommen werden, die durch diese Bezugnahme in den Offenbarungsgehalt der vorliegenden Anmeldung aufgenommen wird. Bei MSC handelt es sich um genormte Ablaufdiagramme, auch Pfeildiagramme oder X-Diagramme genannt. Sie sind unabhängig von Programmierkenntnissen zu verstehen. Eine automatische Ausführung von mit MSC beschriebenen Kommunikationen ist auf Protokolltestern jedoch nicht möglich. Um ausführbare Tests zu erhalten müssen daher sogenannte

Scripts geschrieben werden, was eine Einarbeitung des Anwenders in die jeweilige Programmiersprache erforderlich macht. Eine allgemein verständliche Dokumentation muß zusätzlich erstellt werden. Für einen Test müssen daher getrennt zum einen eine graphische und textuelle Dokumentation erstellt werden, zum anderen ein Source-Code oder ein ausführbarer Binary.

Aus diesem Stand der Technik ergeben sich eine Vielzahl von Nachteilen: Häufig ist eine Konvertierung bestehender Tests nötig, so daß die Gefahr von Inkonsistenzen besteht. Konfigurationsinformationen sind oft in den Spezifikationen der Testkommunikationen nicht oder zumindest nicht maschinenlesbar oder nicht menschenlesbar enthalten. Die verwendeten Sprachen stellen häufig proprietäre Ansätze dar, die von Gerät zu Gerät verschieden sind und neu erlernt werden müssen. Der Anwender wird nicht oder nur rudimentär bei der Erstellung der Nachrichten und Ereignisse mit Protokollwissen unterstützt.

Der vorliegenden Erfindung liegt daher die Aufgabe zugrunde, die Nachteile der oben dargestellten, aus dem Stand der Technik bekannten Vorgehensweise zu überwinden.

Diese Aufgabe wird gelöst durch ein Verfahren gemäß Anspruch 1.

Gemäß einem weiteren Aspekt der Erfindung wird ein Protokolltester zur Verfügung gestellt mit den Merkmalen von Anspruch 8.

Die Erfindung erlaubt es, den gewünschten Kommunikationsablauf in einer leicht verständlichen graphischen, standardisierten Form, beispielsweise MSC, zu erstellen. Durch die erfindungsgemäße Lösung ist es nunmehr möglich, die Kommunikationsschnittstellen und/oder die Kommunikationsdaten graphisch zu spezifizieren. Mit MSC selbst allein ist dies nicht möglich. Dadurch daß den graphisch auswählbaren Parametern Beschreibungsdateien zugeordnet sind, wird die Möglichkeit bereitgestellt, daß der Protokolltester automatisch die vom Benutzer ausgewählten Parameter in eine ausführbare Version eines Kommunikationsablaufs

transformiert. Durch die graphische Bereitstellung von Parameterzusammenstellungen, aus denen der Benutzer auswählen kann, wird der Benutzer in jeder Phase mit konfigurationsspezifischer und protokollspezifischer Information unterstützt. Neben einer kurzen Einweisung in einen Graphikstandard, beispielsweise MSC, benötigt der Benutzer kein weiteres Vorwissen, um das erfindungsgemäße Verfahren erfolgreich anwenden zu können. Die graphische Darstellung erfolgt auf einer Anzeigeeinheit, beispielsweise einem Monitor oder einem Bildschirm.

Dokumentation und Implementierung wird auf diese Weise in einem Vorgang erledigt, wobei der Benutzer immer die Dokumentations-sicht auf einer graphischen Benutzeroberfläche sieht.

Die Bedienungsfreundlichkeit für den Benutzer steigt umso mehr, je mehr Auswahlen dem Benutzer graphisch zur Verfügung gestellt werden. In einer besonders bevorzugten Ausführungsform der Erfindung sind daher, mit Bezug auf das erfindungsgemäße Verfahren, alle Auswahl-schritte gemäß a) bis d) graphisch unterstützt, wobei allen, mit den Auswahlmitteln auswählbaren Parametern Beschreibung-dateien zugeordnet sind, die dann in Schritt e) zur Erstellung eines zwischen den Instanzen ausführbaren Kommunikationsablaufs verwendbar sind.

Die abstrakten Kommunikationsschnittstellen umfassen vorzugsweise sogenannte Service Access Points (SAPs), die Kommunikationsdaten vorzugsweise sogenannte Protocol Data Units (PDUs) und/oder sogenannte Abstract Service Primitives (ASPs). Hierbei handelt es sich um Primitive, d. h. Datenpakete, mit denen sich verschiedene Kommunikationsschichten derselben Instanz untereinander verständigen. Ein SAP ist demnach ein Punkt, an dem verschiedene ASPs ausgetauscht werden können. Vorzugsweise enthalten ASPs PDUs, wobei üblicherweise jede PDU einzeln zu erstellen ist.

Die Auswahl der Kommunikationsdaten kann zwei Teilschritte umfassen, zunächst das graphische Auswählen eines Datenformats,

dann den graphischen Aufbau einer Kommunikationsabfolge zwischen den beteiligten Instanzen.

Hinsichtlich des zuletzt genannten Teilschritts kann die Möglichkeit vorgesehen werden, Source-Code einzugeben.

Die zuvor erwähnten Merkmale der Erfindung gelten in entsprechender Weise für einen erfindungsgemäßen Protokolltester.

Weitere vorteilhafte Weiterbildungen der Erfindung sind in den Unteransprüchen definiert.

Ein Ausführungsbeispiel wird im folgenden, unter Hinweis auf die beigefügten Zeichnungen, näher beschrieben. Es stellen dar:

- Figur 1 eine erste graphische Benutzeroberfläche, wie sie bei dem erfindungsgemäßen Verfahren Anwendung findet;
- Figur 2 eine zweite graphische Benutzeroberfläche, wie sie bei dem erfindungsgemäßen Verfahren Anwendung findet;
- Figur 3 die Benutzeroberfläche von Figur 2 in einem anderen Darstellungsmodus;
- Figur 4 die Benutzeroberfläche von Figur 2 in einem weiteren Darstellungsmodus;
- Figur 5 eine dritte graphische Benutzeroberfläche, wie sie bei dem erfindungsgemäßen Verfahren Anwendung findet;
- Figur 6 die Benutzeroberfläche von Figur 5 in einem anderen Darstellungsmodus; und
- Figur 7 die Benutzeroberfläche von Figur 5 in einem weiteren Darstellungsmodus.

Figur 1 zeigt eine graphische Benutzeroberfläche 10, die in ei-

nem ersten Schritt auf graphischem Wege die Auswahl der an einer Kommunikation beteiligten Instanzen ermöglicht. Graphisches Auswählen im Zusammenhang mit der vorliegenden Erfindung bedeutet, daß ein Symbol oder ein Textvorschlag graphisch auf einer graphischen Benutzeroberfläche, beispielsweise einem PC-Bildschirm, angezeigt wird und durch schlichtes Aktivieren, d. h. beispielsweise durch Anklicken mit einer Maus, ausgewählt werden kann. Eine der Instanzen ist ein Protokolltester, auf dem das erfindungsgemäße Verfahren zur Verfügung gestellt wird, wobei der Protokolltester im vorliegenden Fall eine Komponente TC_1 emuliert. Mit den beiden Icons "Add" 12 und "Delete" 14 kann der Benutzer weitere Instanzen hinzufügen beziehungsweise aufgelistete Instanzen löschen. In einem Feld 16 ist die Zusammenstellung der Instanzen aufgelistet, während sie in einem Feld 18 als Diagramm angezeigt wird. In einem Feld 19 kann der Name der Instanz, in Feld 20 der Typ der Instanz festgelegt werden. Zwei Icons 22, 24 ermöglichen dem Benutzer, sich von einer Stufe der Definition des Kommunikationsablaufs zur nächsten zu bewegen, sowohl in Richtung auf detailliertere Spezifikation sowie auch in Richtung übergeordneter Darstellung. Eine "Cancel"-Taste 26 ermöglicht das Verlassen einer Stufe, wobei die vorgenommenen Änderungen rückgängig gemacht werden. Eine "Help"-Taste 28 bietet dem Benutzer weitere Unterstützung an.

Gemäß Figur 2, in der eine weitere Darstellung der Benutzeroberfläche 10 gezeigt ist, hat der vorliegende Kommunikationsablauf den Namen Gateway_1, siehe Feld 22. An ihm nimmt teil eine erste Instanz TC_1, gemäß Feld 24, sowie eine zweite Instanz IUT_1, gemäß Feld 26. Das emulierte Protokoll ist gemäß Feld 28a vom Typ isdn12, wobei in Feld 28b weitere, zur Auswahl stehende Protokolle angeboten werden. In einem Feld 30 können verschiedene, zur weiteren Bearbeitung auswählbare Kommunikationsabläufe angeboten werden. Die mit Bezug auf Figur 1 beschriebenen Icons 12, 14, 22, 24, 26, 28 finden sich in vergleichbarer Form mit vergleichbarer Funktion wieder, siehe auch die noch folgenden Benutzeroberflächen, und werden daher nicht nochmals beschrieben.

Figur 3 zeigt die Benutzeroberfläche 10 von Figur 1 in einem anderen Darstellungsmodus, und zwar zur Auswahl eines SAPs, siehe Feld 32a. Im Feld 32b werden weitere SAPs zur Auswahl angeboten. Alle in Feld 32b dargestellten SAPs werden zu der gewählten Emulation isdn12 angeboten.

Figur 4 zeigt eine weitere Darstellungsform der Benutzeroberfläche 10 von Figur 2, wobei nunmehr in einem Feld 34 aus sogenannten Message Pools ein Format für die Kommunikationsdaten (ASPs, PDUs) verwendet wird.

Figur 5 zeigt eine weitere Benutzeroberfläche 36, die in einem Feld 38 dem Benutzer diverse Informationen zur Verfügung stellt: Zum einen die von ihm ausgewählten Instanzen, dann das gemäß den Figuren 1 bis 4 vereinbarte Testszenario (Gateway_1) sowie das Datenformat (Message Pools). Im folgenden soll zunächst auf Figur 6 Bezug genommen werden: Die Benutzeroberfläche 36 weist eine Vielzahl von Icons 40 auf, die, wie von Programmen zur Textverarbeitung oder Graphikbearbeitung bekannt, beispielsweise unter Verwendung einer Maus anklickbar sind. Unter Verwendung dieser Icons läßt sich in einem Feld 42 in graphischer Weise ein Kommunikationsablauf erstellen. Figur 6 zeigt die Möglichkeit des Einbaus von Code in der Programmiersprache Forth (Draft Proposal ANSI Standard 1994) in einem Block TE_cfg 44, unter Verwendung einer Eingabemaske 46. Für die Eingabe von Code in einer anderen Programmiersprache können weitere Eingabemasken vorgesehen werden.

Zurück zu Figur 5: Hier wird als Beispiel für einen Abschnitt eines Kommunikationsablaufs in Feld 48 dargestellt, wie zunächst alternativ die ASPs DL_ESTABLISH_CNF oder DL_ESTABLISH_IND von einer Instanz erwartet werden. Daraufhin wird ein Timer T_Waitlnit der Länge 5s gestartet und das Ablaufen des Timers abgewartet.

Figur 7 zeigt, wie eine isdn-PDU "SETUP_1" als Sendenachricht in das graphisch erstellte Ablaufdiagramm eingefügt wird. In einer

Eingabemaske 50 werden ASPs mit PDUs aus dem zuvor ausgewählten Message Pool angeboten. In einem optisch hervorgehobenen Feld 52 kann die ausgewählte PDU eingetragen werden. In einem Feld 54 wird weitere Information zur ausgewählten ASP beziehungsweise PDU dem Benutzer dargeboten.

Auf die soeben beschriebene Art und Weise läßt sich somit ein Kommunikationsablauf erstellen, wobei bevorzugt allen auswählbaren Parametern Beschreibungsdateien zugeordnet sind, die zum automatischen Erstellen eines zwischen den Instanzen ausführbaren Kommunikationsablaufs durch den Protokolltester automatisch miteinander verwendbar sind.

Bei der Erzeugung eines ausführbaren Codes wirken drei Komponenten zusammen: Zunächst die graphischen Benutzeroberflächen, die die gewählten Parameter, insbesondere auch die Kommunikationsabfolge in einer internen Struktur speichert. Dann ein Compiler, der die gewählten Parameter in temporäre Files übersetzt und schließlich ein Linker, der dies temporären Files liest und in die gewählte Interpreter-Scriptsprache, beispielsweise ANS Forth, umsetzt. Dabei wird der gesamte Kommunikationsablauf wie er vom Benutzer definiert wurde in ein Scriptfile geschrieben.

Im Anhang A1 ist passend zu den beschriebenen Figuren der automatisch vom Protokolltester generierte Code wiedergegeben.

Anhang A1

```
( ***** Tektronix MSC-Linker <VO.92.0> builds scenario 'isdn_user' ***-*- forth
--*** )
```

```
" $MSC$Script$" find [if] forget $MSC$Script$ [then] drop variable
$MSC$Script$
" emul" find 0= [if] loadm emul [then] drop
" error" find 0= [if] loadm error [then] drop
" mbslib" find 0= [if] loadm mbslib [then] drop
" mforth" find 0= [if] loadm mforth [then] drop
default-order
v_trace
v_screen
```

```
( >>>>>>>> Allocation <<<<<<<<< )
( create instance variables and constants... )
CREATE $MSC$_InstanceVars 4 ALLOT
```

```
1 CONSTANT MSC_NUM_OF_INSTANCES
$MSC$_InstanceVars MSC_NUM_OF_INSTANCES 4 * 0 FILL
( create timer variables and constants... )
CREATE $MSC$_TimerVars 40 ALLOT
$MSC$_TimerVars 40 0 FILL
CREATE MSC_TIMER 20 ALLOT
MSC_TIMER 20 0 FILL
5 CONSTANT MSC_NUM_OF_TIMERS
( create pool variables and constants... )
CREATE $MSC$_PoolVars 4 ALLOT
1 CONSTANT MSC_NUM_OF_POOLS
$MSC$_PoolVars 4 0 FILL
( create message variables and constants... )
CREATE $MSC$_MsgVars 132 ALLOT
$MSC$_MsgVars 132 0 FILL
11 CONSTANT MSC_NUM_OF_MESSAGES
CREATE $MSC$_MsgDecodeVars 4 ALLOT
$MSC$_MsgDecodeVars 4 0 FILL
1 CONSTANT MSC_NUM_OF_MSGDECODEVARS ( one per TM )
CREATE $MSC$_MsgFolderVars 44 ALLOT
$MSC$_MsgFolderVars 44 0 FILL
11 CONSTANT MSC_NUM_OF_FOLDERS
CREATE $MSC$_EventStructureVars MSC_NUM_OF_POOLS MSC_NUM_OF_INSTANCES * 4 *
ALLOT
$MSC$_EventStructureVars MSC_NUM_OF_POOLS MSC_NUM_OF_INSTANCES * 4 * 0 FILL
CREATE $MSC$_MsgSizeVars 4 ALLOT
$MSC$_MsgSizeVars 4 0 FILL
variable $MSC$_MsgMatched?
( create temporary variables and constants... )
variable $MSC$_TempFolderHandle
variable $MSC$_PDecoutVar
( create startstate variables... )
```

```
variable $MSC$Req-State
```

```
( >>>>>>> Constants <<<<<<<<< )
```

```
( create mapping of gateway name to poolindex )
```

```
0 constant MSC-GW-Gateway_1 \ Mapping Gatewayname 'Gateway_1' -> Poolindex '0'
```

```
( >>>>>>> Variables <<<<<<<<< )
```

```
variable MSC-VAR-Gateway_1-SAPI
```

```
variable MSC-VAR-Gateway_1-TEI
```

```
( >>>>>>> Commands <<<<<<<<< )
```

```
include pc:boot:/share/pfe/msc_lib.4th
```

```
( constructor word ... )
```

```
: $MSC$Constructor ( -- )
```

```
    " Cannot open pool 'pc:c:/k1297/MBS-Pools/STK-etsi93-pool1.pdc' " "
pc:c:/k1297/MBS-Pools/STK-etsi93-pool1.pdc" 0 $MSC$OpenPool \ open pool
'pc:c:/k1297/MBS-Pools/STK-etsi93-pool1.pdc'
```

```
    " Cannot open folder 'PROT<ETSI> send to EMUL<isdnl2>' " " PROT<ETSI> send
to EMUL<isdnl2>" 0 8 $MSC$OpenFolder \ open folder 'PROT<ETSI> send to
EMUL<isdnl2>' within pool 'pc:c:/k1297/MBS-Pools/STK-etsi93-pool1.pdc'
```

```
    " Cannot init message 'rCONN_1' " " rCONN_1" 0 8 8 $MSC$InitMsgVar \ init
message 'rCONN_1' of pool 'pc:c:/k1297/MBS-Pools/STK-etsi93-pool1.pdc'
```

```
    " Cannot open folder 'PROT<ETSI> send to EMUL<isdnl2>' " " PROT<ETSI> send
to EMUL<isdnl2>" 0 3 $MSC$OpenFolder \ open folder 'PROT<ETSI> send to
EMUL<isdnl2>' within pool 'pc:c:/k1297/MBS-Pools/STK-etsi93-pool1.pdc'
```

```
    " Cannot init message 'rDL_ESTABLISH_CNF_1' " " rDL_ESTABLISH_CNF_1" 0 3 3
$MSC$InitMsgVar \ init message 'rDL_ESTABLISH_CNF_1' of pool
'pc:c:/k1297/MBS-Pools/STK-etsi93-pool1.pdc'
```

```
    " Cannot open folder 'PROT<ETSI> send to EMUL<isdnl2>' " " PROT<ETSI> send
to EMUL<isdnl2>" 0 2 $MSC$OpenFolder \ open folder 'PROT<ETSI> send to
EMUL<isdnl2>' within pool 'pc:c:/k1297/MBS-Pools/STK-etsi93-pool1.pdc'
```

```
    " Cannot init message 'sDL_ESTABLISH_REQ_1' " " sDL_ESTABLISH_REQ_1" 0 2 2
$MSC$InitMsgVar \ init message 'sDL_ESTABLISH_REQ_1' of pool
'pc:c:/k1297/MBS-Pools/STK-etsi93-pool1.pdc'
```

```
    " Cannot open folder 'PROT<ETSI> send to EMUL<isdnl2>' " " PROT<ETSI> send
to EMUL<isdnl2>" 0 4 $MSC$OpenFolder \ open folder 'PROT<ETSI> send to
EMUL<isdnl2>' within pool 'pc:c:/k1297/MBS-Pools/STK-etsi93-pool1.pdc'
```

```
    " Cannot init message 'rDL_ESTABLISH_IND_1' " " rDL_ESTABLISH_IND_1" 0 4 4
$MSC$InitMsgVar \ init message 'rDL_ESTABLISH_IND_1' of pool
'pc:c:/k1297/MBS-Pools/STK-etsi93-pool1.pdc'
```

```
    " Cannot open folder 'PROT<ETSI> send to EMUL<isdnl2>' " " PROT<ETSI> send
to EMUL<isdnl2>" 0 10 $MSC$OpenFolder \ open folder 'PROT<ETSI> send to
EMUL<isdnl2>' within pool 'pc:c:/k1297/MBS-Pools/STK-etsi93-pool1.pdc'
```

```
    " Cannot init message 'rREL_COM_1' " " rREL_COM_1" 0 10 10 $MSC$InitMsgVar
\ init message 'rREL_COM_1' of pool 'pc:c:/k1297/MBS-Pools/STK-etsi93-
pool1.pdc'
```

```
    " Cannot open folder 'PROT<ETSI> send to EMUL<isdnl2>' " " PROT<ETSI> send
to EMUL<isdnl2>" 0 7 $MSC$OpenFolder \ open folder 'PROT<ETSI> send to
EMUL<isdnl2>' within pool 'pc:c:/k1297/MBS-Pools/STK-etsi93-pool1.pdc'
```

```
    " Cannot init message 'rALERT_1' " " rALERT_1" 0 7 7 $MSC$InitMsgVar \
```

```

init message 'rALERT_1' of pool 'pc:c:/k1297/MBS-Pools/STK-etsi93-pool1.pdc'
    " Cannot open folder 'PROT<ETSI> send to EMUL<isdnl2>' " " PROT<ETSI> send
to EMUL<isdnl2>" 0 9 $MSC$_OpenFolder \ open folder 'PROT<ETSI> send to
EMUL<isdnl2>' within pool 'pc:c:/k1297/MBS-Pools/STK-etsi93-pool1.pdc'
    " Cannot init message 'sDISC_1' " " sDISC_1" 0 9 9 $MSC$_InitMsgVar \ init
message 'sDISC_1' of pool 'pc:c:/k1297/MBS-Pools/STK-etsi93-pool1.pdc'
    " Cannot open folder 'PROT<ETSI> send to EMUL<isdnl2>' " " PROT<ETSI> send
to EMUL<isdnl2>" 0 6 $MSC$_OpenFolder \ open folder 'PROT<ETSI> send to
EMUL<isdnl2>' within pool 'pc:c:/k1297/MBS-Pools/STK-etsi93-pool1.pdc'
    " Cannot init message 'rCALL_PROC_1' " " rCALL_PROC_1" 0 6 6
$MSC$_InitMsgVar \ init message 'rCALL_PROC_1' of pool 'pc:c:/k1297/MBS-
Pools/STK-etsi93-pool1.pdc'
    " Cannot open folder 'PROT<ETSI> send to EMUL<isdnl2>' " " PROT<ETSI> send
to EMUL<isdnl2>" 0 1 $MSC$_OpenFolder \ open folder 'PROT<ETSI> send to
EMUL<isdnl2>' within pool 'pc:c:/k1297/MBS-Pools/STK-etsi93-pool1.pdc'
    " Cannot init message 'rMDL_ASSIGN_CNF_1' " " rMDL_ASSIGN_CNF_1" 0 1 1
$MSC$_InitMsgVar \ init message 'rMDL_ASSIGN_CNF_1' of pool 'pc:c:/k1297/MBS-
Pools/STK-etsi93-pool1.pdc'
    " Cannot open folder 'PROT<ETSI> send to EMUL<isdnl2>' " " PROT<ETSI> send
to EMUL<isdnl2>" 0 0 $MSC$_OpenFolder \ open folder 'PROT<ETSI> send to
EMUL<isdnl2>' within pool 'pc:c:/k1297/MBS-Pools/STK-etsi93-pool1.pdc'
    " Cannot init message 'sMDL_ASSIGN_REQ_1' " " sMDL_ASSIGN_REQ_1" 0 0 0
$MSC$_InitMsgVar \ init message 'sMDL_ASSIGN_REQ_1' of pool 'pc:c:/k1297/MBS-
Pools/STK-etsi93-pool1.pdc'
    " Cannot open folder 'PROT<ETSI> send to EMUL<isdnl2>' " " PROT<ETSI> send
to EMUL<isdnl2>" 0 5 $MSC$_OpenFolder \ open folder 'PROT<ETSI> send to
EMUL<isdnl2>' within pool 'pc:c:/k1297/MBS-Pools/STK-etsi93-pool1.pdc'

    " Cannot init message 'sSETUP_1' " " sSETUP_1" 0 5 5 $MSC$_InitMsgVar \
init message 'sSETUP_1' of pool 'pc:c:/k1297/MBS-Pools/STK-etsi93-pool1.pdc'
    MSC-VAR-Gateway_1-SAPI " SAPI" " PROT<ETSI> send to EMUL<isdnl2>" 0
$MSC$_AssignMSCVar
    MSC-VAR-Gateway_1-TEI " TEI" " PROT<ETSI> send to EMUL<isdnl2>" 0
$MSC$_AssignMSCVar
;

( destructor word ... )
: $MSC$_Destructor ( -- )
    1 0 DO
        I $MSC$_GetPoolHandle k12mbspoolclose DROP
    LOOP
;

( >>>>>>>>> Initialization <<<<<<<<<< )
5000 4 $MSC$_SetExtTimerVar \ init. timer 'T_Pause' of instance 'Phone'
45000 1 $MSC$_SetExtTimerVar \ init. timer 'T310' of instance 'Phone'
4000 0 $MSC$_SetExtTimerVar \ init. timer 'T303' of instance 'Phone'
30000 3 $MSC$_SetExtTimerVar \ init. timer 'T305' of instance 'Phone'
20000 2 $MSC$_SetExtTimerVar \ init. timer 'T_Call' of instance 'Phone'
0 0 $MSC$_InitMsg \ Create k12MBSevent structure for instance 'Phone' and
gateway 'Gateway_1'

```

TM0 (>>>>>>>> start of instance 'Phone' <<<<<<<<<)

(Segments of Instance 'Phone':

Type	Segment Name	State	Length
INIT	- no name -	0000000000	0000000001
END	- no name -	0000000001	0000000001
DOC	START	0000000002	0000000004
DOC	NULL	0000000006	0000000001
DOC	CALL_INITIATED	0000000007	0000000003
DOC	CALL_PROCEEDING	0000000010	0000000003
DOC	CALL_DELIVERED_ACTIV	0000000013	0000000002
DOC	DISCONNECT_REQUEST	0000000015	0000000004
CONN	NULL	0000000019	0000000001
CONN	CALL_INITIATED	0000000020	0000000001
CONN	CALL_PROCEEDING	0000000021	0000000001
CONN	CALL_DELIVERED_ACTIV	0000000022	0000000001

\ ----- init segment -----

```

0 STATE_INIT{
    128 " TM0 starts" $MSC$_TraceMsg
    0 $MSC$_ResetGotoModifierFlag \ init. instance 'Phone'
    4 $MSC$_InitTimerVar \ init. timer 'T_Pause'
    1 $MSC$_InitTimerVar \ init. timer 'T310'
    0 $MSC$_InitTimerVar \ init. timer 'T303'
    3 $MSC$_InitTimerVar \ init. timer 'T305'
    2 $MSC$_InitTimerVar \ init. timer 'T_Call'
    ( switch command for startstate... )
    $MSC$_Req-State @ CASE
        1 OF 2 NEW_STATE ENDOF

        2 OF 6 NEW_STATE ENDOF
        2 0 $MSC$_NewState ( goto START )
    ENDCASE
}STATE_INIT

```

\ ----- end segment -----

```

1 STATE_INIT{
    " instance 'Phone' stops" $MSC$_PrintString
    128 " TM0 stops" $MSC$_TraceMsg
}STATE_INIT
1 STATE{
    ( this is the end state - loop forever )
}STATE

```

\ ----- document segment 'START' -----

```

2 STATE_INIT{
    32 " Forthbox TE_cfg start " $MSC$_TraceMsg
    ( start forth box 'TE_cfg' ) " config lapd.General.Side=TE_PM"

```

```

EMU_ADMIN ( end forth box 'TE_cfg' )
    64 " Forthbox TE_cfg end " $MSC$_TraceMsg
    16 " Send message 'PROT<ETSI> send to EMUL<isdnl2>/sMDL_ASSIGN_REQ_1'
over gateway 'Gateway_1 ' " $MSC$_TraceMsg
    " Cannot send message 'sMDL_ASSIGN_REQ_1'" 0 0 0 $MSC$_SendPrimitive
}STATE_INIT
2 STATE{
    " Error while matching primitive 'rMDL_ASSIGN_CNF_1'" 1 0 0
$MSC$_RecvPrimitive
    ACTION{
        8 " Received message 'PROT<ETSI> send to
EMUL<isdnl2>/rMDL_ASSIGN_CNF_1' from gateway 'Gateway_1 ' " $MSC$_TraceMsg
        0 0 1 $MSC$_FreeEventStructure \ free event structure of message
'PROT<ETSI> send to EMUL<isdnl2>/rMDL_ASSIGN_CNF_1' and gateway 'Gateway_1'
        1 $MSC$_ResetMsgFlag \ message 'PROT<ETSI> send to
EMUL<isdnl2>/rMDL_ASSIGN_CNF_1' from gateway 'Gateway_1'
        0 $MSC$_ResetGotoModifierFlag
        16 " Send message 'PROT<ETSI> send to
EMUL<isdnl2>/sDL_ESTABLISH_REQ_1' over gateway 'Gateway_1 ' " $MSC$_TraceMsg
        " Cannot send message 'sDL_ESTABLISH_REQ_1'" 2 0 0
$MSC$_SendPrimitive
        3 0 $MSC$_NewState
    }ACTION
    ?TM_TIMEOUT
    ACTION{
        1 " Unexpected timer event " $MSC$_TraceMsg
    }ACTION
    FALSE E-SAP @ 0 = OR
    ACTION{
        8 " Unexpected message event " $MSC$_TraceMsg
        0 0 1 $MSC$_FreeEventStructure \ free event structure of message
'PROT<ETSI> send to EMUL<isdnl2>/rMDL_ASSIGN_CNF_1' and gateway 'Gateway_1'
    }ACTION
}STATE
3 STATE_INIT{
    3 $MSC$_ResetMsgFlag \ message 'PROT<ETSI> send to
EMUL<isdnl2>/rDL_ESTABLISH_CNF_1' from gateway 'Gateway_1'

    4 $MSC$_ResetMsgFlag \ message 'PROT<ETSI> send to
EMUL<isdnl2>/rDL_ESTABLISH_IND_1' from gateway 'Gateway_1'
}STATE_INIT
3 STATE{
    " Error while matching primitive 'rDL_ESTABLISH_CNF_1'" 3 0 0
$MSC$_RecvPrimitive
    ACTION{
        0 0 3 $MSC$_FreeEventStructure \ free event structure of message
'PROT<ETSI> send to EMUL<isdnl2>/rDL_ESTABLISH_CNF_1' and gateway 'Gateway_1'
        0 0 4 $MSC$_FreeEventStructure \ free event structure of message
'PROT<ETSI> send to EMUL<isdnl2>/rDL_ESTABLISH_IND_1' and gateway 'Gateway_1'
        0 $MSC$_SetGotoModifierFlag
        4 0 $MSC$_NewState
    }ACTION
}STATE

```

```

}ACTION
  " Error while matching primitive 'rDL_ESTABLISH_IND_1'" 4 0 0
$MSC$_RecvPrimitive
  ACTION{
    0 0 3 $MSC$_FreeEventStructure \ free event structure of message
    'PROT<ETSI> send to EMUL<isdnl2>/rDL_ESTABLISH_CNF_1' and gateway 'Gateway_1'
    0 0 4 $MSC$_FreeEventStructure \ free event structure of message
    'PROT<ETSI> send to EMUL<isdnl2>/rDL_ESTABLISH_IND_1' and gateway 'Gateway_1'
    0 $MSC$_SetGotoModifierFlag
    5 0 $MSC$_NewState
  }ACTION
  ?TM_TIMEOUT
  ACTION{
    1 " Unexpected timer event " $MSC$_TraceMsg
    0 0 3 $MSC$_FreeEventStructure \ free event structure of message
    'PROT<ETSI> send to EMUL<isdnl2>/rDL_ESTABLISH_CNF_1' and gateway 'Gateway_1'
    0 0 4 $MSC$_FreeEventStructure \ free event structure of message
    'PROT<ETSI> send to EMUL<isdnl2>/rDL_ESTABLISH_IND_1' and gateway 'Gateway_1'
  }ACTION
  FALSE E-SAP @ 0 = OR
  ACTION{
    8 " Unexpected message event " $MSC$_TraceMsg
    0 0 3 $MSC$_FreeEventStructure \ free event structure of message
    'PROT<ETSI> send to EMUL<isdnl2>/rDL_ESTABLISH_CNF_1' and gateway 'Gateway_1'
    0 0 4 $MSC$_FreeEventStructure \ free event structure of message
    'PROT<ETSI> send to EMUL<isdnl2>/rDL_ESTABLISH_IND_1' and gateway 'Gateway_1'
  }ACTION
}STATE
4 STATE{
  " Error while matching primitive 'rDL_ESTABLISH_CNF_1'" 3 0 0
$MSC$_RecvPrimitive
  ACTION{
    8 " Received message 'PROT<ETSI> send to
    EMUL<isdnl2>/rDL_ESTABLISH_CNF_1' from gateway 'Gateway_1 ' " $MSC$_TraceMsg
    0 0 3 $MSC$_FreeEventStructure \ free event structure of message
    'PROT<ETSI> send to EMUL<isdnl2>/rDL_ESTABLISH_CNF_1' and gateway 'Gateway_1'
    3 $MSC$_ResetMsgFlag \ message 'PROT<ETSI> send to
    EMUL<isdnl2>/rDL_ESTABLISH_CNF_1' from gateway 'Gateway_1'
    0 $MSC$_ResetGotoModifierFlag
    6 0 $MSC$_NewState
  }ACTION
  ?TM_TIMEOUT
  ACTION{
    1 " Unexpected timer event " $MSC$_TraceMsg
  }ACTION
  FALSE E-SAP @ 0 = OR
  ACTION{
    8 " unexpected message event " $MSC$_TraceMsg
    0 0 3 $MSC$_FreeEventStructure \ free event structure of message
    'PROT<ETSI> send to EMUL<isdnl2>/rDL_ESTABLISH_CNF_1' and gateway 'Gateway_1'
  }ACTION
}STATE
5 STATE{

```



```
" Error while matching primitive 'rDL_ESTABLISH_IND_1'" 4 0 0
$MSC$_RecvPrimitive
  ACTION{
    8 " Received message 'PROT<ETSI> send to
EMUL<isdnl2>/rDL_ESTABLISH_IND_1' from gateway 'Gateway_1 ' " $MSC$_TraceMsg
    0 0 4 $MSC$_FreeEventStructure \ free event structure of message
'PROT<ETSI> send to EMUL<isdnl2>/rDL_ESTABLISH_IND_1' and gateway 'Gateway_1'
    4 $MSC$_ResetMsgFlag \ message 'PROT<ETSI> send to
EMUL<isdnl2>/rDL_ESTABLISH_IND_1' from gateway 'Gateway_1'
    0 $MSC$_ResetGotoModifierFlag
    6 0 $MSC$_NewState
  }ACTION
  ?TM_TIMEOUT
  ACTION{
    1 " Unexpected timer event " $MSC$_TraceMsg
  }ACTION
  FALSE E-SAP @ 0 = OR
  ACTION{
    8 " Unexpected message event " $MSC$_TraceMsg
    0 0 4 $MSC$_FreeEventStructure \ free event structure of message
'PROT<ETSI> send to EMUL<isdnl2>/rDL_ESTABLISH_IND_1' and gateway 'Gateway_1'
  }ACTION
}STATE

\ ----- document segment 'NULL' -----
6 STATE_INIT{
  16 " Send message 'PROT<ETSI> send to EMUL<isdnl2>/sSETUP_1' over
gateway 'Gateway_1 ' " $MSC$_TraceMsg
  " Cannot send message 'sSETUP_1'" 5 0 0 $MSC$_SendPrimitive
  2 " Timer 'T303' set with value '4000'" $MSC$_TraceMsg
  4000 0 $MSC$_SetTimer \ timer 'T303'
  19 0 $MSC$_NewState
}STATE_INIT

\ ----- document segment 'CALL_INITIATED' -----
7 STATE{
  " Error while matching primitive 'rCALL_PROC_1'" 6 0 0
$MSC$_RecvPrimitive
  ACTION{
    0 0 6 $MSC$_FreeEventStructure \ free event structure of message
'PROT<ETSI> send to EMUL<isdnl2>/rCALL_PROC_1' and gateway 'Gateway_1'
    0 $MSC$_SetGotoModifierFlag
    8 0 $MSC$_NewState
  }ACTION
  0 $MSC$_Timeout \ timer 'T303'
  ACTION{
    0 0 6 $MSC$_FreeEventStructure \ free event structure of message
'PROT<ETSI> send to EMUL<isdnl2>/rCALL_PROC_1' and gateway 'Gateway_1'
    0 $MSC$_SetGotoModifierFlag
    9 0 $MSC$_NewState
```

```
}ACTION
?TM_TIMEOUT
ACTION{
    1 " Unexpected timer event " $MSC$TraceMsg
    0 0 6 $MSC$FreeEventStructure \ free event structure of message
'PROT<ETSI> send to EMUL<isdnl2>/rCALL_PROC_1' and gateway 'Gateway_1'
}ACTION
FALSE E-SAP @ 0 = OR
ACTION{
    8 " Unexpected message event " $MSC$TraceMsg
    0 0 6 $MSC$FreeEventStructure \ free event structure of message
'PROT<ETSI> send to EMUL<isdnl2>/rCALL_PROC_1' and gateway 'Gateway_1'
}ACTION
}STATE
8 STATE{
    " Error while matching primitive 'rCALL_PROC_1'" 6 0 0
$MSC$RecvPrimitive
    ACTION{
        8 " Received message 'PROT<ETSI> send to EMUL<isdnl2>/rCALL_PROC_1'
from gateway 'Gateway_1' " $MSC$TraceMsg
        0 0 6 $MSC$FreeEventStructure \ free event structure of message
'PROT<ETSI> send to EMUL<isdnl2>/rCALL_PROC_1' and gateway 'Gateway_1'
        6 $MSC$ResetMsgFlag \ message 'PROT<ETSI> send to
EMUL<isdnl2>/rCALL_PROC_1' from gateway 'Gateway_1'
        0 $MSC$ResetGotoModifierFlag
        4 " Timer 'T303' reset" $MSC$TraceMsg
        0 $MSC$ResetTimer \ timer 'T303'
        2 " Timer 'T310' set with value '45000'" $MSC$TraceMsg
        45000 1 $MSC$SetTimer \ timer 'T310'
        20 0 $MSC$NewState
    }ACTION
    ?TM_TIMEOUT
    ACTION{
        1 " Unexpected timer event " $MSC$TraceMsg
    }ACTION
    FALSE E-SAP @ 0 = OR
    ACTION{
        8 " Unexpected message event " $MSC$TraceMsg
        0 0 6 $MSC$FreeEventStructure \ free event structure of message
'PROT<ETSI> send to EMUL<isdnl2>/rCALL_PROC_1' and gateway 'Gateway_1'
    }ACTION
    }STATE
9 STATE{
    0 $MSC$Timeout \ timer 'T303'
    ACTION{
        1 " Received timeout 'T303' " $MSC$TraceMsg
        0 $MSC$ResetTimerFlag \ timer 'T303'
        0 $MSC$ResetGotoModifierFlag
        20 0 $MSC$NewState
    }ACTION
    ?TM_TIMEOUT
    ACTION{
```

```

        1 " Unexpected timer event " $MSC$_TraceMsg
    }ACTION
    FALSE E-SAP @ 0 = OR
    ACTION{
        8 " Unexpected message event " $MSC$_TraceMsg
    }ACTION
}STATE

\ ----- document segment 'CALL_PROCEEDING' -----
10 STATE{
    " Error while matching primitive 'rALERT_1' " 7 0 0 $MSC$_RecvPrimitive
    ACTION{
        0 0 7 $MSC$_FreeEventStructure \ free event structure of message
        'PROT<ETSI> send to EMUL<isdnl2>/rALERT_1' and gateway 'Gateway_1'
        0 $MSC$_SetGotoModifierFlag
        11 0 $MSC$_NewState
    }ACTION
    1 $MSC$_Timeout \ timer 'T310'
    ACTION{
        0 0 7 $MSC$_FreeEventStructure \ free event structure of message
        'PROT<ETSI> send to EMUL<isdnl2>/rALERT_1' and gateway 'Gateway_1'
        0 $MSC$_SetGotoModifierFlag
        12 0 $MSC$_NewState
    }ACTION
    ?TM_TIMEOUT
    ACTION{
        1 " Unexpected timer event " $MSC$_TraceMsg
        0 0 7 $MSC$_FreeEventStructure \ free event structure of message
        'PROT<ETSI> send to EMUL<isdnl2>/rALERT_1' and gateway 'Gateway_1'
    }ACTION
    FALSE E-SAP @ 0 = OR
    ACTION{
        8 " Unexpected message event " $MSC$_TraceMsg
        0 0 7 $MSC$_FreeEventStructure \ free event structure of message
        'PROT<ETSI> send to EMUL<isdnl2>/rALERT_1' and gateway 'Gateway_1'
    }ACTION
}STATE
11 STATE{
    " Error while matching primitive 'rALERT_1' " 7 0 0 $MSC$_RecvPrimitive
    ACTION{
        8 " Received message 'PROT<ETSI> send to EMUL<isdnl2>/rALERT_1' from
gateway 'Gateway_1 ' " $MSC$_TraceMsg
        0 0 7 $MSC$_FreeEventStructure \ free event structure of message
        'PROT<ETSI> send to EMUL<isdnl2>/rALERT_1' and gateway 'Gateway_1'
        7 $MSC$_ResetMsgFlag \ message 'PROT<ETSI> send to
EMUL<isdnl2>/rALERT_1' from gateway 'Gateway_1'
        0 $MSC$_ResetGotoModifierFlag
        4 " Timer 'T310' reset" $MSC$_TraceMsg
        1 $MSC$_ResetTimer \ timer 'T310'
        21 0 $MSC$_NewState
    }ACTION
    ?TM_TIMEOUT

```

```

ACTION{
    1 " Unexpected timer event " $MSC$_TraceMsg
}ACTION
FALSE E-SAP @ 0 = OR

ACTION{
    8 " Unexpected message event " $MSC$_TraceMsg
    0 0 7 $MSC$_FreeEventStructure \ free event structure of message
'PROT<ETSI> send to EMUL<isdnl2>/rALERT_1' and gateway 'Gateway_1'
}ACTION
}STATE
12 STATE{
    1 $MSC$_Timeout \ timer 'T310'
    ACTION{
        1 " Received timeout 'T310' " $MSC$_TraceMsg
        1 $MSC$_ResetTimerFlag \ timer 'T310'
        0 $MSC$_ResetGotoModifierFlag
        21 0 $MSC$_NewState
    }ACTION
    ?TM_TIMEOUT
    ACTION{
        1 " Unexpected timer event " $MSC$_TraceMsg
    }ACTION
    FALSE E-SAP @ 0 = OR
    ACTION{
        8 " Unexpected message event " $MSC$_TraceMsg
    }ACTION
}STATE

\ ----- document segment 'CALL_DELIVERED_ACTIVE' -----
13 STATE{
    " Error while matching primitive 'rCONN_1'" 8 0 0 $MSC$_RecvPrimitive
    ACTION{
        8 " Received message 'PROT<ETSI> send to EMUL<isdnl2>/rCONN_1' from
gateway 'Gateway_1 ' " $MSC$_TraceMsg
        0 0 8 $MSC$_FreeEventStructure \ free event structure of message
'PROT<ETSI> send to EMUL<isdnl2>/rCONN_1' and gateway 'Gateway_1'
        8 $MSC$_ResetMsgFlag \ message 'PROT<ETSI> send to
EMUL<isdnl2>/rCONN_1' from gateway 'Gateway_1'
        0 $MSC$_ResetGotoModifierFlag
        2 " Timer 'T_Call' set with value '20000'" $MSC$_TraceMsg
        20000 2 $MSC$_SetTimer \ timer 'T_Call'
        14 0 $MSC$_NewState
    }ACTION
    ?TM_TIMEOUT
    ACTION{
        1 " Unexpected timer event " $MSC$_TraceMsg
    }ACTION
    FALSE E-SAP @ 0 = OR
    ACTION{
        8 " Unexpected message event " $MSC$_TraceMsg
    }ACTION
}STATE

```

```

    0 0 8 $MSC$_FreeEventStructure \ free event structure of message
'PROT<ETSI> send to EMUL<isdnl2>/rCONN_1' and gateway 'Gateway_1'
    }ACTION
    }STATE
14 STATE{
    2 $MSC$_Timeout \ timer 'T_Call'
    ACTION{
        1 " Received timeout 'T_Call' " $MSC$_TraceMsg
        2 $MSC$_ResetTimerFlag \ timer 'T_Call'
        0 $MSC$_ResetGotoModifierFlag

        16 " Send message 'PROT<ETSI> send to EMUL<isdnl2>/sDISC_1' over
gateway 'Gateway_1 ' " $MSC$_TraceMsg
        " Cannot send message 'sDISC_1'" 9 0 0 $MSC$_SendPrimitive
        2 " Timer 'T305' set with value '30000'" $MSC$_TraceMsg
        30000 3 $MSC$_SetTimer \ timer 'T305'
        22 0 $MSC$_NewState
    }ACTION
    ?TM_TIMEOUT
    ACTION{
        1 " Unexpected timer event " $MSC$_TraceMsg
    }ACTION
    FALSE E-SAP @ 0 = OR
    ACTION{
        8 " Unexpected message event " $MSC$_TraceMsg
    }ACTION
    }STATE

\ ----- document segment 'DISCONNECT_REQUEST' -----
15 STATE{
    " Error while matching primitive 'rREL_COM_1'" 10 0 0
$MSC$_RecvPrimitive
    ACTION{
        0 0 10 $MSC$_FreeEventStructure \ free event structure of message
'PROT<ETSI> send to EMUL<isdnl2>/rREL_COM_1' and gateway 'Gateway_1'
        0 $MSC$_SetGotoModifierFlag
        16 0 $MSC$_NewState
    }ACTION
    3 $MSC$_Timeout \ timer 'T305'
    ACTION{
        0 0 10 $MSC$_FreeEventStructure \ free event structure of message
'PROT<ETSI> send to EMUL<isdnl2>/rREL_COM_1' and gateway 'Gateway_1'
        0 $MSC$_SetGotoModifierFlag
        17 0 $MSC$_NewState
    }ACTION
    ?TM_TIMEOUT
    ACTION{
        1 " Unexpected timer event " $MSC$_TraceMsg
        0 0 10 $MSC$_FreeEventStructure \ free event structure of message
'PROT<ETSI> send to EMUL<isdnl2>/rREL_COM_1' and gateway 'Gateway_1'
    }ACTION
    FALSE E-SAP @ 0 = OR

```

```
ACTION{
    8 " Unexpected message event " $MSC$_TraceMsg
    0 0 10 $MSC$_FreeEventStructure \ free event structure of message
'PROT<ETSI> send to EMUL<isdnl2>/rREL_COM_1' and gateway 'Gateway_1'
}ACTION
}STATE
16 STATE{
    " Error while matching primitive 'rREL_COM_1'" 10 0 0
$MSC$_RecvPrimitive
    ACTION{
        8 " Received message 'PROT<ETSI> send to EMUL<isdnl2>/rREL_COM_1'
from gateway: 'Gateway_1 ' " $MSC$_TraceMsg
        0 0 10 $MSC$_FreeEventStructure \ free event structure of message
'PROT<ETSI> send to EMUL<isdnl2>/rREL_COM_1' and gateway 'Gateway_1'

        10 $MSC$_ResetMsgFlag \ message 'PROT<ETSI> send to
EMUL<isdnl2>/rREL_COM_1' from gateway 'Gateway_1'
        0 $MSC$_ResetGotoModifierFlag
        18 0 $MSC$_NewState
    }ACTION
    ?TM_TIMEOUT
    ACTION{
        1 " Unexpected timer event " $MSC$_TraceMsg
    }ACTION
    FALSE E-SAP @ 0 = OR
    ACTION{
        8 " Unexpected message event " $MSC$_TraceMsg
        0 0 10 $MSC$_FreeEventStructure \ free event structure of message
'PROT<ETSI> send to EMUL<isdnl2>/rREL_COM_1' and gateway 'Gateway_1'
    }ACTION
}STATE
17 STATE{
    3 $MSC$_Timeout \ timer 'T305'
    ACTION{
        1 " Received timeout 'T305' " $MSC$_TraceMsg
        3 $MSC$_ResetTimerFlag \ timer 'T305'
        0 $MSC$_ResetGotoModifierFlag
        18 0 $MSC$_NewState
    }ACTION
    ?TM_TIMEOUT
    ACTION{
        1 " Unexpected timer event " $MSC$_TraceMsg
    }ACTION
    FALSE E-SAP @ 0 = OR
    ACTION{
        8 " Unexpected message event " $MSC$_TraceMsg
    }ACTION
}STATE
18 STATE_INIT{
    2 " Timer 'T_Pause' set with value '5000'" $MSC$_TraceMsg
    5000 4 $MSC$_SetTimer \ timer 'T_Pause'
```

```
}STATE_INIT
18 STATE{
    4 $MSC$_Timeout \ timer 'T_Pause'
    ACTION{
        1 " Received timeout 'T_Pause' " $MSC$_TraceMsg
        4 $MSC$_ResetTimerFlag \ timer 'T_Pause'
        0 $MSC$_ResetGotoModifierFlag
        1 0 $MSC$_NewState
    }ACTION
    ?TM_TIMEOUT
    ACTION{
        1 " Unexpected timer event " $MSC$_TraceMsg
    }ACTION
    FALSE E-SAP @ 0 = OR
    ACTION{
        8 " Unexpected message event " $MSC$_TraceMsg
    }ACTION
}STATE

\ ----- connector segment 'NULL' -----

19 STATE_INIT{
    6 $MSC$_ResetMsgFlag \ message 'PROT<ETSI> send to
EMUL<isdnl2>/rCALL_PROC_1'
    0 $MSC$_ResetTimerFlag \ timer 'T303'
}STATE_INIT
19 STATE{
    " Error while matching primitive 'rCALL_PROC_1'" 6 0 0
$MSC$_RecvPrimitive
    ACTION{
        0 0 6 $MSC$_FreeEventStructure \ free event structure of message
'PROT<ETSI> send to EMUL<isdnl2>/rCALL_PROC_1' and gateway 'Gateway_1'
        0 $MSC$_SetGotoModifierFlag
        6 $MSC$_SetMsgFlag \ message 'PROT<ETSI> send to
EMUL<isdnl2>/rCALL_PROC_1'
        7 0 $MSC$_NewState
    }ACTION
    0 $MSC$_Timeout \ timer 'T303'
    ACTION{
        0 0 6 $MSC$_FreeEventStructure \ free event structure of message
'PROT<ETSI> send to EMUL<isdnl2>/rCALL_PROC_1' and gateway 'Gateway_1'
        0 $MSC$_SetGotoModifierFlag
        0 $MSC$_SetTimerFlag \ timer 'T303'
        7 0 $MSC$_NewState
    }ACTION
    ?TM_TIMEOUT
    ACTION{
        " Unexpected timer event" $MSC$_TraceMsg
    }ACTION
    FALSE E-SAP @ 0 = OR
    ACTION{
        " Unexpected message event" $MSC$_TraceMsg
```

```

        0 0 6 $MSC$FreeEventStructure \ free event structure of message
'PROT<ETSI> send to EMUL<isdnl2>/rCALL_PROC_1' and gateway 'Gateway_1'
    }ACTION
}STATE

\ ----- connector segment 'CALL_INITIATED' -----
20 STATE_INIT{
    7 $MSC$ResetMsgFlag \ message 'PROT<ETSI> send to
EMUL<isdnl2>/rALERT_1'
    1 $MSC$ResetTimerFlag \ timer 'T310'
}STATE_INIT
20 STATE{
    " Error while matching primitive 'rALERT_1'" 7 0 0 $MSC$RecvPrimitive
    ACTION{
        0 0 7 $MSC$FreeEventStructure \ free event structure of message
'PROT<ETSI> send to EMUL<isdnl2>/rALERT_1' and gateway 'Gateway_1'
        0 $MSC$SetGotoModifierFlag
        7 $MSC$SetMsgFlag \ message 'PROT<ETSI> send to
EMUL<isdnl2>/rALERT_1'
        10 0 $MSC$NewState
    }ACTION
    1 $MSC$Timeout \ timer 'T310'
    ACTION{
        0 0 7 $MSC$FreeEventStructure \ free event structure of message
'PROT<ETSI> send to EMUL<isdnl2>/rALERT_1' and gateway 'Gateway_1'

        0 $MSC$SetGotoModifierFlag
        1 $MSC$SetTimerFlag \ timer 'T310'
        10 0 $MSC$NewState
    }ACTION
    ?TM_TIMEOUT
    ACTION{
        " Unexpected timer event" $MSC$TraceMsg
    }ACTION
    FALSE E-SAP @ 0 = OR
    ACTION{
        " Unexpected message event" $MSC$TraceMsg
        0 0 7 $MSC$FreeEventStructure \ free event structure of message
'PROT<ETSI> send to EMUL<isdnl2>/rALERT_1' and gateway 'Gateway_1'
    }ACTION
}STATE

\ ----- connector segment 'CALL_PROCEEDING' -----
21 STATE_INIT{
    8 $MSC$ResetMsgFlag \ message 'PROT<ETSI> send to
EMUL<isdnl2>/rCONN_1'
}STATE_INIT
21 STATE{
    " Error while matching primitive 'rCONN_1'" 8 0 0 $MSC$RecvPrimitive
    ACTION{
        0 0 8 $MSC$FreeEventStructure \ free event structure of message
'PROT<ETSI> send to EMUL<isdnl2>/rCONN_1' and gateway 'Gateway_1'
    }

```



```

    0 $MSC$_SetGotoModifierFlag
    8 $MSC$_SetMsgFlag \ message 'PROT<ETSI> send to
EMUL<isdnl2>/rCONN_1'
    13 0 $MSC$_NewState
}ACTION
?TM_TIMEOUT
ACTION{
    " Unexpected timer event" $MSC$_TraceMsg
}ACTION
FALSE E-SAP @ 0 = OR
ACTION{
    " Unexpected message event" $MSC$_TraceMsg
    0 0 8 $MSC$_FreeEventStructure \ free event structure of message
'PROT<ETSI> send to EMUL<isdnl2>/rCONN_1' and gateway 'Gateway_1'
}ACTION
}STATE

\ ----- connector segment 'CALL_DELIVERED_ACTIVE' -----
22 STATE_INIT{
    10 $MSC$_ResetMsgFlag \ message 'PROT<ETSI> send to
EMUL<isdnl2>/rREL_COM_1'
    3 $MSC$_ResetTimerFlag \ timer 'T305'
}STATE_INIT
22 STATE{
    " Error while matching primitive 'rREL_COM_1'" 10 0 0
$MSC$_RecvPrimitive
    ACTION{
        0 0 10 $MSC$_FreeEventStructure \ free event structure of message
'PROT<ETSI> send to EMUL<isdnl2>/rREL_COM_1' and gateway 'Gateway_1'
        0 $MSC$_SetGotoModifierFlag
    }
    10 $MSC$_SetMsgFlag \ message 'PROT<ETSI> send to
EMUL<isdnl2>/rREL_COM_1'
    15 0 $MSC$_NewState
}ACTION
3 $MSC$_Timeout \ timer 'T305'
ACTION{
    0 0 10 $MSC$_FreeEventStructure \ free event structure of message
'PROT<ETSI> send to EMUL<isdnl2>/rREL_COM_1' and gateway 'Gateway_1'
    0 $MSC$_SetGotoModifierFlag
    3 $MSC$_SetTimerFlag \ timer 'T305'
    15 0 $MSC$_NewState
}ACTION
?TM_TIMEOUT
ACTION{
    " Unexpected timer event" $MSC$_TraceMsg
}ACTION
FALSE E-SAP @ 0 = OR
ACTION{
    " Unexpected message event" $MSC$_TraceMsg
    0 0 10 $MSC$_FreeEventStructure \ free event structure of message
'PROT<ETSI> send to EMUL<isdnl2>/rREL_COM_1' and gateway 'Gateway_1'
}
```

```
        }ACTION
    }STATE
( >>>>>>>> end of instance 'Phone' <<<<<<<<< )

$MSC$_Constructor
MSC_MENU_CTRL_FCT ( calls the menu control function )
" MSC scenario 'isdn_user' loaded" $MSC$_PrintString
```

Tektronix, Inc.
Europäische Patentneuanmeldung

Anwaltsakte: 25380
23 Feb 2000

**Verfahren zum Erstellen eines Ablaufs einer zwischen
mindestens zwei Instanzen ablaufenden Kommunikation und dieses
Verfahren realisierender Protokolltester**

Patentansprüche:

1. Verfahren zum Erstellen eines Ablaufs einer zwischen mindestens zwei Instanzen ablaufenden Kommunikation, wobei eine Instanz ein Protokolltester ist, gekennzeichnet durch folgende, am Protokolltester ausführbare Schritte:
 - a) Auswählen der an der Kommunikation beteiligten Instanzen;
 - b) Auswählen einer Protokollschicht, auf deren Grundlage die Kommunikation zwischen den ausgewählten Instanzen ablaufen soll;
 - c) Auswählen derjenigen abstrakten Kommunikationsschnittstellen der Protokollschicht, die an der Kommunikation beteiligt sind;
 - d) Auswählen der Kommunikationsdaten;
 - e) automatisches Erstellen eines zwischen den mindestens zwei Instanzen ausführbaren Kommunikationsablaufs durch den Protokolltester, auf der Grundlage der Auswahlen in den Schritten a) bis d),wobei die Auswahl von Schritt c) und/oder Schritt d) graphisch erfolgt und den dabei auswählbaren Parametern Beschreibungsdateien zugeordnet sind, die in Schritt e) zur Erstellung eines zwischen den Instanzen ausführbaren Kommunikationsablaufs verwendet werden.
2. Verfahren nach Anspruch 1,
dadurch gekennzeichnet,

daß weiterhin in Schritt a) die an der Kommunikation beteiligten Instanzen graphisch ausgewählt werden und/oder in Schritt b) die Protokollschicht graphisch ausgewählt wird und den dabei auswählbaren Parametern Beschreibungsdateien zugeordnet sind, die in Schritt e) zur Erstellung eines zwischen den Instanzen ausführbaren Kommunikationsablaufs verwendet werden.

3. Verfahren nach Anspruch 1 oder 2,
dadurch gekennzeichnet,
daß die abstrakten Kommunikationsschnittstellen SAPs (Service Access Points) umfassen.
4. Verfahren nach einem der vorhergehenden Ansprüche,
dadurch gekennzeichnet,
daß die Kommunikationsdaten PDUs (Protocol Data Units) und/oder ASPs (Abstract Service Primitives) umfassen.
5. Verfahren nach einem der vorhergehenden Ansprüche,
dadurch gekennzeichnet,
daß Schritt d) folgende Teilschritte umfaßt:
d1) graphisches Auswählen eines Datenformats;
d2) graphischer Aufbau einer Kommunikationsabfolge zwischen den beteiligten Instanzen.
6. Verfahren nach Anspruch 5,
dadurch gekennzeichnet,
daß in Schritt d2) Source-Code eingebbar ist.
7. Verfahren nach einem der vorhergehenden Ansprüche,
dadurch gekennzeichnet,
daß allen auswählbaren Parametern Beschreibungsdateien zugeordnet sind, die in Schritt e) zur Erstellung eines zwischen den Instanzen ausführbaren Kommunikationsablaufs verwendet werden.
8. Protokolltester mit

- a) Mitteln zum Auswählen der an einer Kommunikation beteiligten Instanzen (19, 20, 24, 26), wobei eine der Instanzen der Protokolltester ist;
- b) Mitteln zum Auswählen einer Protokollschicht (20, 28a), auf deren Grundlage die Kommunikation zwischen den ausgewählten Instanzen ablaufen soll;
- c) Mitteln zum Auswählen derjenigen abstrakten Kommunikationsschnittstellen (20, 32a) der Protokollschicht, die an der Kommunikation beteiligt sind;
- d) Mitteln zum Auswählen der Kommunikationsdaten (20, 34);
- e) Mitteln zum automatischen Erstellen eines zwischen den Instanzen ausführbaren Kommunikationsablaufs durch den Protokolltester, auf der Grundlage der Auswahlen gemäß a) bis d),

wobei die Auswahlmittel gemäß c) und/oder gemäß d) graphische Auswahlmittel sind und den durch sie auswählbaren Parametern Beschreibungsdateien zugeordnet sind, die gemäß e) von den Erstellungsmitteln zur Erstellung eines zwischen den Instanzen ausführbaren Kommunikationsablaufs verwendbar sind.

9. Protokolltester nach Anspruch 8,

dadurch gekennzeichnet,

daß die Mittel zum Auswählen der an der Kommunikation beteiligten Instanzen (19, 20, 24, 26) und/oder die Mittel zum Auswählen der Protokollschicht (20, 28a) graphische Auswahlmittel sind und den durch sie auswählbaren Parametern Beschreibungsdateien zugeordnet sind, die gemäß e) von den Erstellungsmitteln zur Erstellung eines zwischen den Instanzen ausführbaren Kommunikationsablaufs verwendbar sind.

10. Protokolltester nach einem der Ansprüche 8 oder 9,

dadurch gekennzeichnet,

daß die abstrakten Kommunikationsschnittstellen SAPs (Service Access Points) umfassen.

11. Protokolltester nach einem der Ansprüche 8 bis 10,
dadurch gekennzeichnet,
daß die Kommunikationsdaten PDUs (Protocol Data Units)
und/oder ASPs (Abstract Service Primitives) umfassen.
12. Protokolltester nach einem der Ansprüche 8 bis 11,
dadurch gekennzeichnet,
daß er Mittel zum Eingeben von Source-Code (44, 46) umfaßt.
13. Protokolltester nach einem der Ansprüche 8 bis 12,
dadurch gekennzeichnet,
daß allen mit den Auswahlmitteln auswählbaren Parametern
Beschreibungsdateien zugeordnet sind, die gemäß e) von den
Erstellungsmitteln zur Erstellung eines zwischen den In-
stanzen ausführbaren Kommunikationsablaufs verwendbar sind.

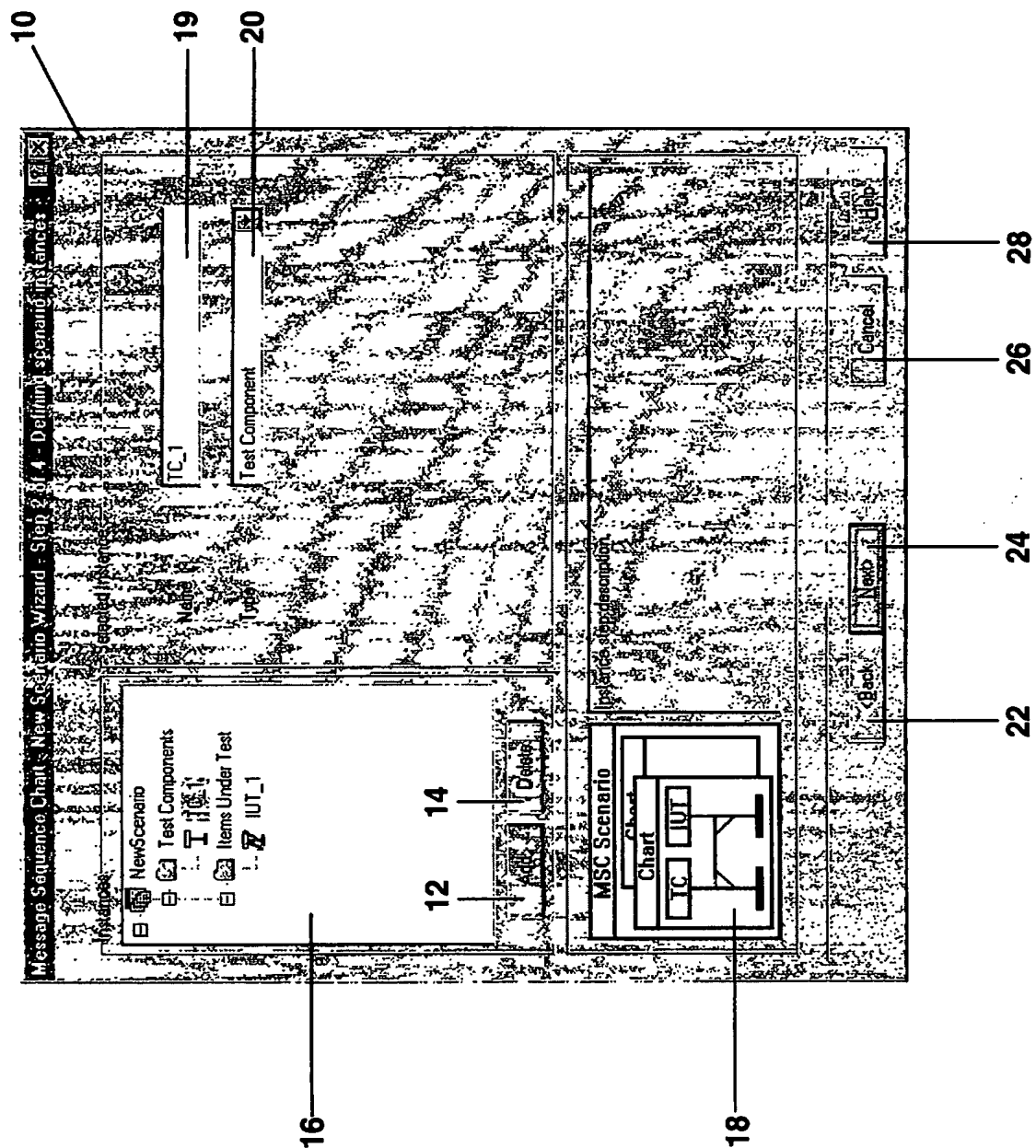
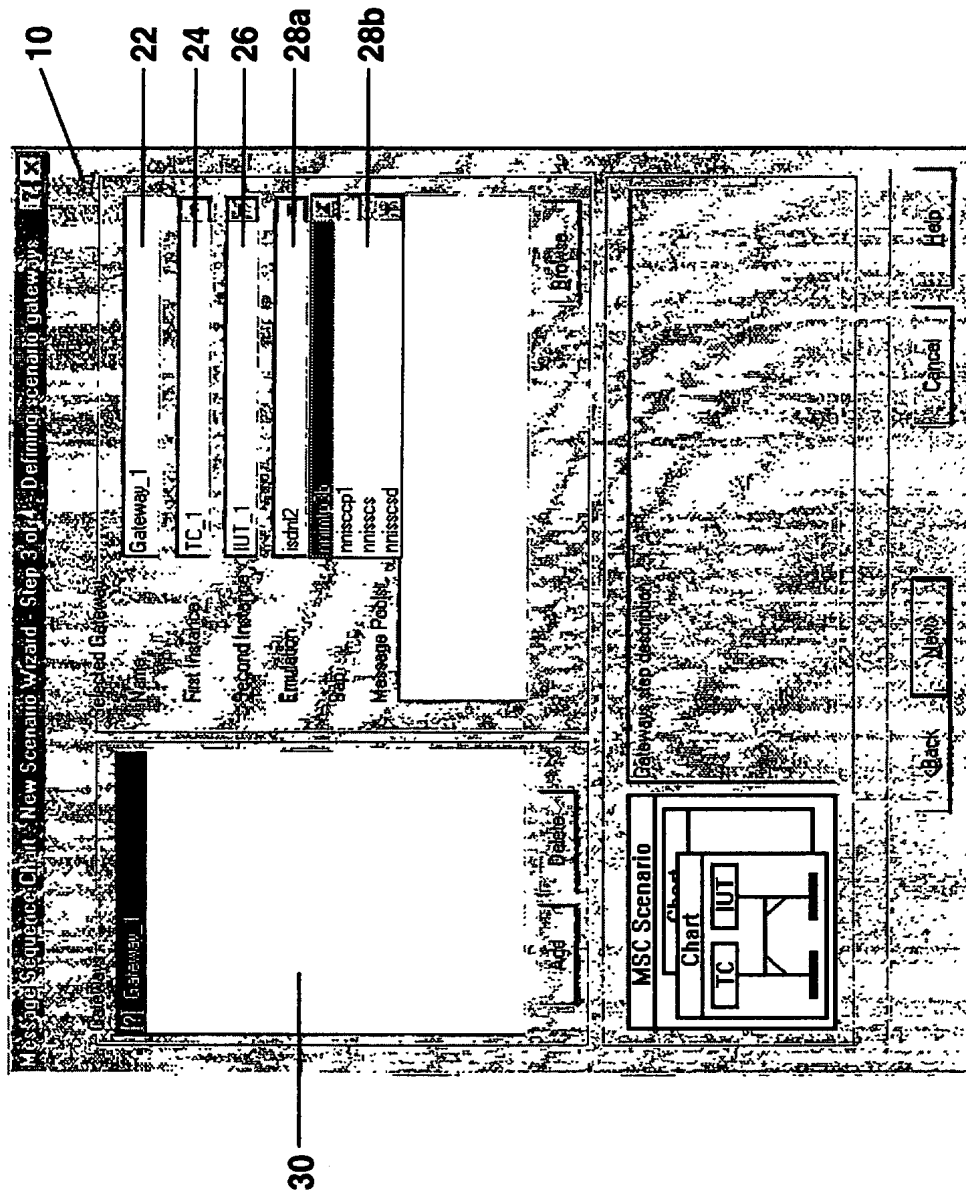


Fig. 1



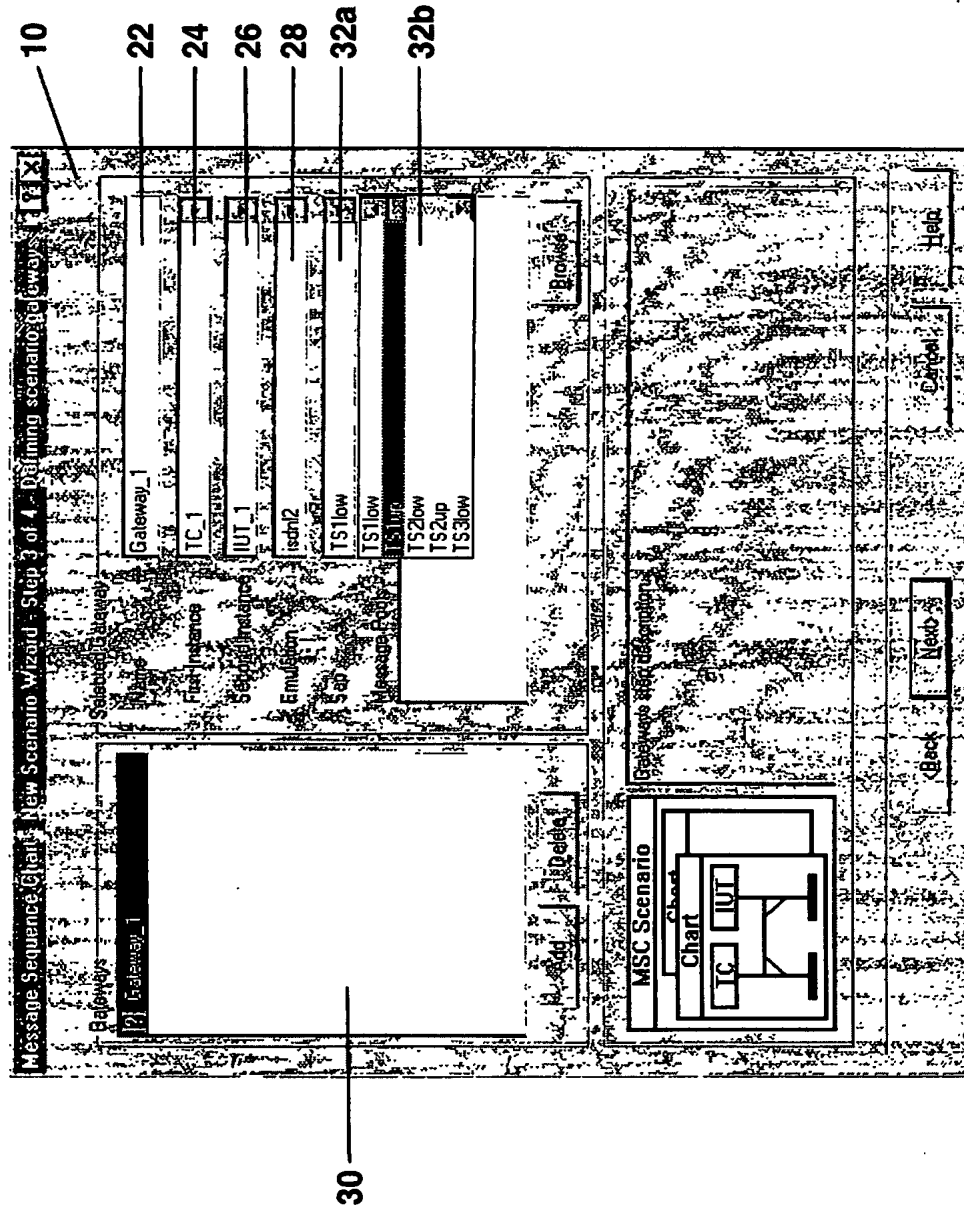


Fig. 3

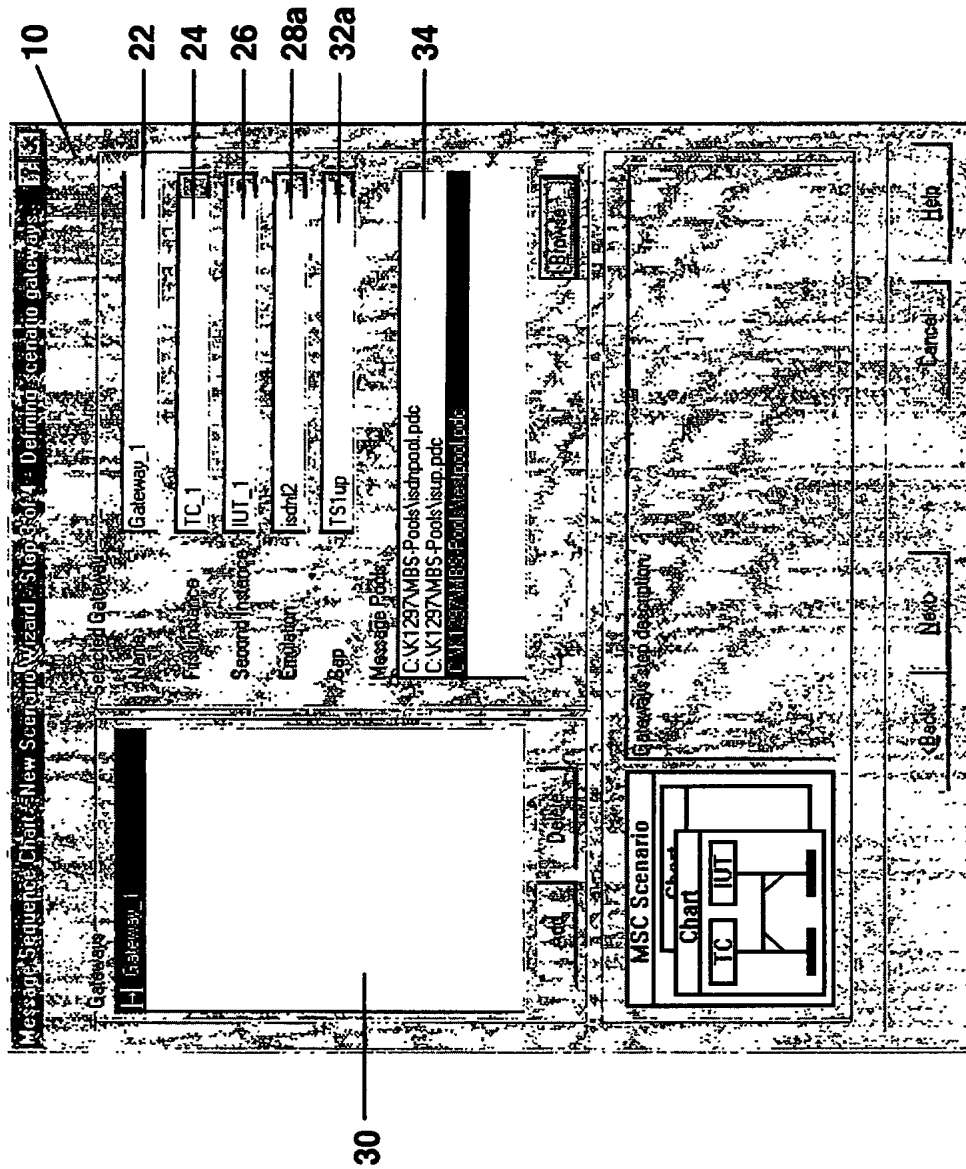


Fig. 4

36

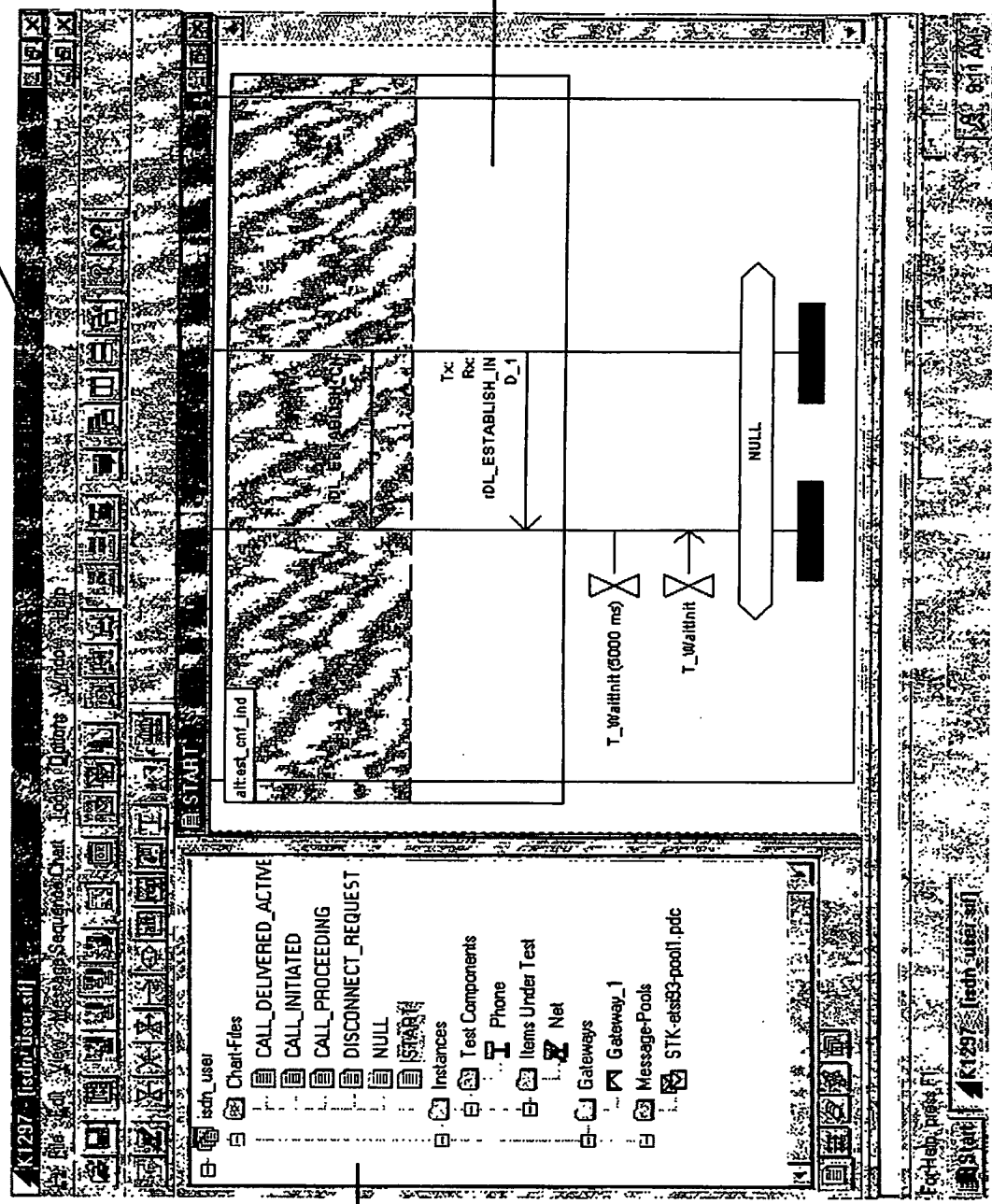
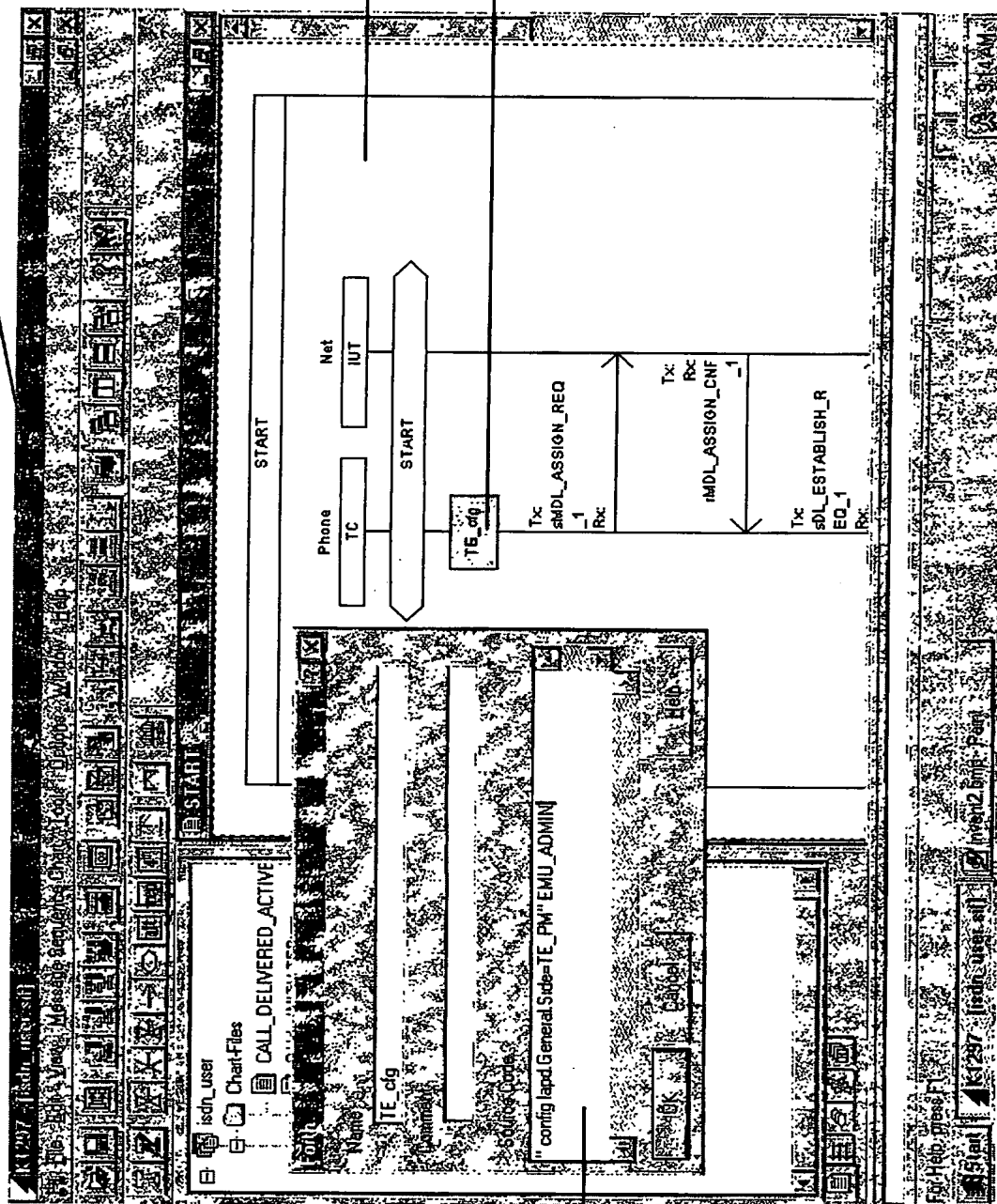


Fig. 5

36



6/7

Fig. 6

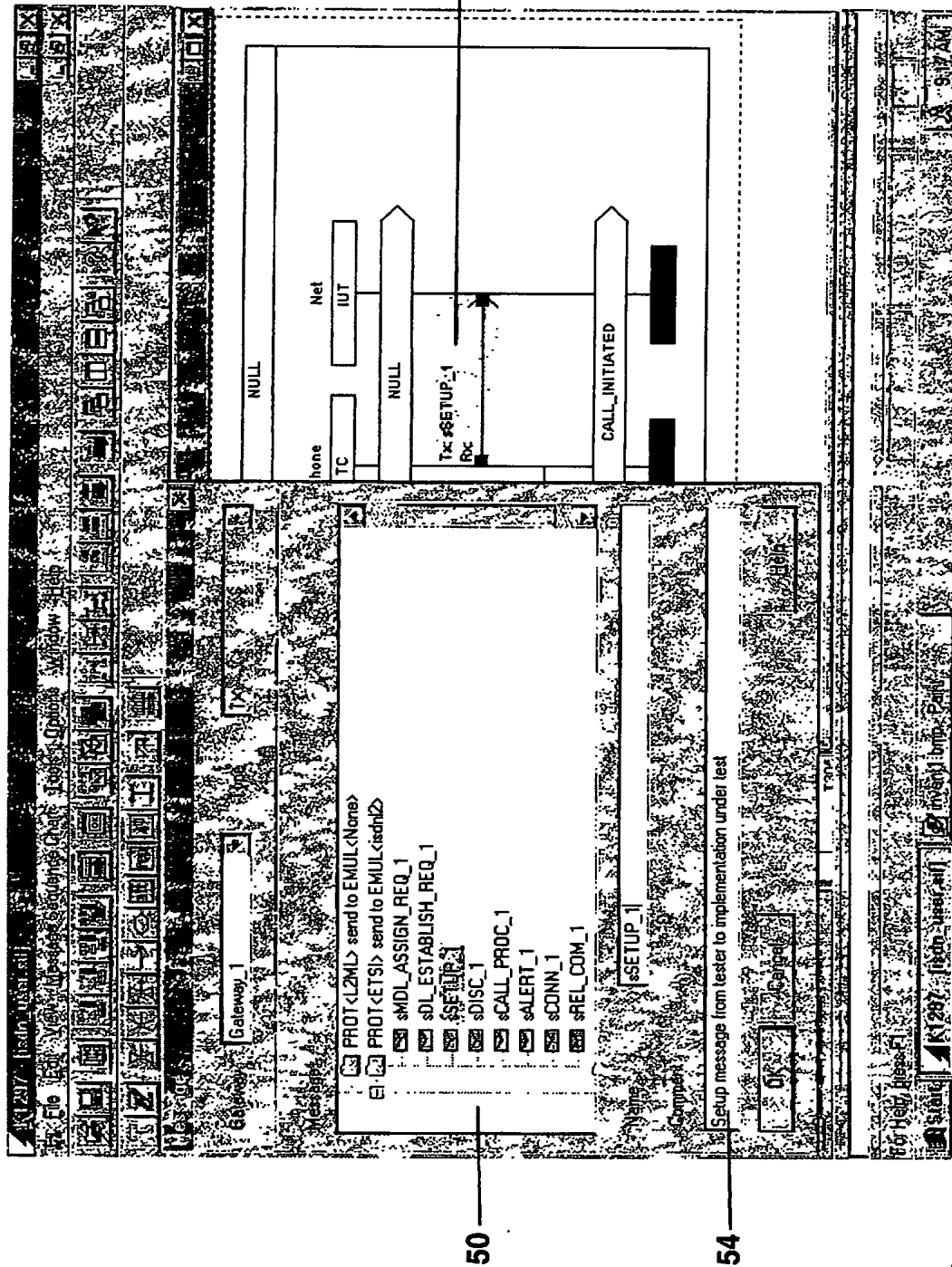


Fig. 7

23. Feb. 2000

- 28 -

Tektronix, Inc.
Europäische Patentneuanmeldung

Anwaltsakte: 25380

**Verfahren zum Erstellen eines Ablaufs einer zwischen
mindestens zwei Instanzen ablaufenden Kommunikation und dieses
Verfahren realisierender Protokolltester**

Zusammenfassung:

Die vorliegende Erfindung betrifft ein Verfahren zum Erstellen eines Ablaufs einer zwischen mindestens zwei Instanzen ablaufenden Kommunikation, wobei eine Instanz ein Protokolltester ist, gekennzeichnet durch folgende, am Protokolltester ausführbare Schritte: a) Auswählen der an der Kommunikation beteiligten Instanzen; b) Auswählen einer Protokollschicht, auf deren Grundlage die Kommunikation zwischen den ausgewählten Instanzen ablaufen soll; c) Auswählen derjenigen abstrakten Kommunikationsschnittstellen der Protokollschicht, die an der Kommunikation beteiligt sind; d) Auswählen der Kommunikationsdaten; e) automatisches Erstellen eines zwischen den mindestens zwei Instanzen ausführbaren Kommunikationsablaufs durch den Protokolltester, auf der Grundlage der Auswahlen in den Schritten a) bis d), wobei die Auswahl von Schritt c) und/oder Schritt d) graphisch erfolgt und den dabei auswählbaren Parametern Beschreibungsdateien zugeordnet sind, die in Schritt e) zur Erstellung eines zwischen den Instanzen ausführbaren Kommunikationsablaufs verwendet werden. Die Erfindung betrifft weiterhin einen Protokolltester, in dem das erfindungsgemäße Verfahren realisiert ist.

(Fig. 6)

Figur für die Zusammenfassung:

Fig. 6

